FMCDP2.cpp

2004年3月4日

全6ページ

岩村覚三

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>
#include <sys¥timeb.h>

//FMCDP2.cpp
//Fuzzy Metric Clustering and Dynamic Programming by
//岩村　覚三、　堀口　正之、　蔵野　正美　2004年3月4日

#define MMAX 24
#define NMAX 25
#define ldmax 100
#define stsize 10

//プロトタイプ宣言//
float dijal(int N, float alpha, int ld, float *adni, float *aupi, float adnj[],
float aupj[]);
float q(float alpha, int ld, float adnj[], float adni[], float p[]);
int Makeup_eval(int N, int k, float eval[], float d[][NMAX], int ks);
float evaluate(int N, int k, float d[][NMAX], int w[]);
int Subsets(int N, int k, int tbl[], int ks);
int Prepare_Network(int M, int N, unsigned int NS[], short int min[], short int
max[]);

int main(void)
{
    char *tzstr = "TZ = JST-9";
    short int min[MMAX + 1], max[MMAX + 1];
    int i, j, al, ks, kk, M, N, ld, k, K, is, ds, rds, dummy,stateKm1k;// rds re
medy ds .
    int **state, **trace_back;
    char pbname[101];
    float **dpv, adn[NMAX][ldmax], aup[NMAX][ldmax], d[NMAX][NMAX], eval[3355443
2];
    unsigned int NS[MMAX + 1];

    //ファイル読み込み//
    FILE *in, *out, *ft;
    in = fopen("InCluster.txt","r");
    out = fopen("OutCluster.txt","w");
    ft = fopen("Ctime.txt","w");

    fscanf(in,"%s",pbname);
    fscanf(in,"%d %d %d",&M,&N,&ld);

    //************************************//
    struct timeb st,et,tt;
    putenv(tzstr);
    tzset();
    ftime(&st);
    //************************************//

    rds = pow(2,N) - 1;

    for(k = 0;k < ld;k++)
        {
            fscanf(in,"%d",&dummy);
            for(i = 0;i < N;i++)
                {
                    fscanf(in,"%f %f",&adn[i][k],&aup[i][k]);
//                  fprintf(out,"ld = %d k = %d i = %d %f %f¥n",ld,k,i,adn[i][k]
,aup[i][k]);//Debug
                }
        }
    fprintf(out,"%s N = %d M = %d¥n",pbname,N,M);
    fprintf(out,"¥n¥n");
    fprintf(out,"alpha     partition     ¥n¥n");

    if((1 > ld) || (ld > ldmax))
        {
            printf("Error No 510");
            return 510;
        }
    if((2 > M) || (M > N) || (N > NMAX))
        {
            printf("Error No 520");
            return 520;
        }

    for(K = 0;K <= M;K++)
        NS[K] = 1;

    printf("stage1 ");
    dummy = Prepare_Network(M, N, NS, min, max);
    //debag//
    //printf("min[1] = %d max[1] = %d¥n min[2] = %d max[2] = %d¥n",min[1],max[1]
,min[2],max[2]);
    ////////
    if(dummy)
```

```
        {
            printf("Error No 530");
            return 530;
        }
    NS[0] = 1;
    NS[M] = 1;
    printf("......clear¥n");
    printf("stage2 ");
    //ダイナミックアロケーション//
    state = new int*[M + 1];
    trace_back = new int*[M + 1];
    dpv = new float*[M + 1];

    for(K = 0;K <= M;K++)
        {
            state[K] = new int[NS[K]];

            if(!state[K])
                {
                    printf("Allocation Error of state[%d].520",K);
                }

        }
   for(K = 0;K <= M;K++)
        {
            trace_back[K] = new int[NS[K]];

            if(!trace_back[K])
                {
                    printf("Allocation Error of trace_back[%d].520",K);
                }

        }
for(K = 0;K <= M;K++)
        {

            dpv[K] = new float[NS[K]];

            if(!dpv[K])
                {
                    printf("Allocation Error of dpv[%d].530",K);
                }
        }


    printf("......clear¥n");

    printf("stage3 ");
    //states生成//
    state[0][0] = 0;
    for(K = 1;K <= M;K++)
        {
            ks = 0;
            for(k = min[K];k <= max[K];k++)//Caution! k >= 2 ?
                ks = Subsets(N, k, state[K], ks);
        }

    printf("......clear¥n");
    printf("stage4 ");
    dpv[0][0] = 0.0;
    for(al = 1;al <= stsize;al++)
        {
            //if(al % 20 == 0)//
                printf(".%d",al);

            for(j = 1;j < N;j++)
                {
                for(i = 0;i < j;i++)
                    {
                        d[i][j] = dijal(N, float(al) / float(stsize), ld, adn[i]
, aup[i], adn[j], aup[j]);
                        d[j][i] = d[i][j];
                        //if(al == 1)fprintf(out," d[%d][%d] = %f",i,j,d[i][j]);
//Debug
                    }
                    //if(al == 1)fprintf(out,"¥n");//Debug
                }

            for(kk = 0;kk < 33554432;kk++)
                eval[kk] = 0.0;

            kk = 0;
            for(k = 0;k <= max[1];k++)
                kk = Makeup_eval(N, k, eval, d, kk);

            for(K = 1;K <= M;K++)
                {
                    for(kk = 0;kk < NS[K];kk++)
                        {
                            dpv[K][kk] = HUGE_VAL;
                            for(k = 0;k < NS[K - 1];k++)
```

```
                                {
                                    stateKm1k = state[K - 1][k];
                                    //printf("k = %d K = %d kk = %d¥n",k,K,kk);/
/debug write
                                    //printf("state[K - 1][kk] = %d state[K][kk]
 = %d¥n",state[K - 1][kk],state[K][kk]);
                                    is = (stateKm1k & state[K][kk]);
                                    ds = (state[K][kk]) ^ (stateKm1k);
                                    ds = (ds & rds);// rds remedy ds .
                                    //printf("ds = %d¥n",ds);
                                    if((is == stateKm1k) && (ds != 0))
                                        {
                                            if(dpv[K][kk] > (dpv[K - 1][k] + eva
l[ds]))
                                                {
                                                    dpv[K][kk] = dpv[K - 1][k] +
 eval[ds];
                                                    trace_back[K][kk] = k;
                                                }
                                        }
                                }
                        }
                }
            fprintf(out,"alpha = %f", al / float(stsize));
            //printf("alpha = %f¥n", al / float(stsize));//Debug write
            fprintf(out," min. DP value = %f",dpv[M][0]);
            fprintf(out," partition = ");

            kk = 0;
            for(K = M;K >= 1;K--)
                {
                    ds = (state[K][kk] ^ state[K - 1][trace_back[K][kk]]);
                    ds = (ds & rds);// rds remedy ds .
                    for(k = N - 1;k >= 0;k--)
                        if((ds << (31 - k)) < 0)
                            fprintf(out,"%d ",k);

                    kk = trace_back[K][kk];
                    fprintf(out,"  ");
                }
            fprintf(out,"¥n¥n");
            /*
            //Debug write start//
            if(al == 1){
                fprintf(out,"¥n¥ntrace_back[k][i] state[k][i] dpv[k][i]¥n");
                for(k = 1;k <= M;k++)
                    {
                        for(i = 0;i < NS[k];i++)
                            fprintf(out,"k = %2d i = %3d %4d %4d %f ¥n",k,i,trac
e_back[k][i],state[k][i],dpv[k][i]);
                            fprintf(out,"¥n¥n¥n");
                    }
                fprintf(out,"¥n¥nal = %d evals are : ¥n",al);
                for(i = 0;i < pow(2,N);i++)
                    {
                        if(i % 10 == 0)
                        fprintf(out,"¥n");

                        fprintf(out," i = %d %f",i,eval[i]);
                    }
            }
            fprintf(out,"¥n¥n¥n");
            //Debug write end//
            */
        }
//      for(K = 0;K <= M;K++)//MS visualC++ .NETではこのdelete命令を入れるとエラ
ーが起こるが、Turbo C++ 4.0Jでは必要。
//          {
//              delete[] state[K];
//              delete[] trace_back[K];
//              delete[] dpv[K];
//          }
    printf(".clear¥n");
    printf("completed¥n¥n");

    //*****************************************//
    ftime(&et);
    tt.time = et.time - st.time;
    //*****************************************//
    fprintf(ft,"tt.time(sec) = %d",tt.time);

    fclose(in);
    fclose(out);
    return (0);
}
float dijal(int N, float alpha, int ld, float *adni, float *aupi, float adnj[],
float aupj[])
{
    int k;
    float p[ldmax], Qalpha, p2, alphaup;
```

```
    for(k = 0;k < ld;k++)
        p[k] = adnj[k] - adni[k] + aupi[k] - aupj[k];

    p2 = 0.0;
    for(k = 0;k < ld;k++)
        p2 += p[k] * p[k];

    alphaup = 0.0;
    for(k = 0;k < ld;k++)
        alphaup += p[k] * (adni[k] - adnj[k]);

    alphaup = -alphaup / p2;
    if(alphaup <= 0.0)
        {
            Qalpha = q(alpha, ld, adnj, adni, p);
        }
    else if(alphaup >= 0.5)
        {
            Qalpha = q(0.0, ld, adnj, adni, p);
        }
    else if(alpha <= 2.0 * alphaup)
        {
            Qalpha = q(0.0, ld, adnj, adni, p);
        }
    else
        {
            Qalpha = q(alpha, ld, adnj, adni, p);
        }
    return sqrt(Qalpha);
}
float q(float alpha, int ld, float adnj[], float adni[], float p[])
{
    int k;
    float qalpha;

    qalpha = 0.0;
    for(k = 0;k < ld;k++)
        qalpha += (p[k] * alpha + adni[k] - adnj[k]) * (p[k] * alpha + adni[k] -
 adnj[k]);

    return qalpha;
}
int Makeup_eval(int N, int k, float eval[], float d[][NMAX], int ks)
{
    int w[25], end[25];
    int kk, i, IntOne = 1;
    int s, bt;

//  for(kk = 1;kk < N;kk++)
//      for(i = 0;i < N;i++)
//          printf("Makeup_eval: d[%d][%d] = %f ¥n",i,kk,d[i][kk]);//Debug


    if(!((2 <= N) && (N <= 25) && (2 <= k) && (k < N)))
        return 610;

    kk = ks;
    for(i = 0;i < k;i++)
        {
            w[i] = i;
            end[i] = N - k + 1 + i;
        }

    s = 0;
    for(i = 0;i < k;i++)
        s = s | (IntOne << w[i]);

//  printf("k = %d s at Makeup_eval 1    s= %d¥n",k,s);
    eval[s] = evaluate(N, k, d, w);

    top1:

    bt = k - 1;

    top2:

    kk++;
    w[bt]++;
    if(w[bt] < N)
        {
            s = 0;
            for(i = 0;i < k;i++)
                s = s | (IntOne << w[i]);
            //printf("k = %d s at Makeup_eval 2    s= %d¥n",k,s);
            eval[s] = evaluate(N, k, d, w);
            goto top2;
        }

    top3:

    bt--;
```

```
    w[bt]++;

    if(w[bt] < end[bt])
        {
            for(i = bt + 1;i < k;i++)
                w[i] = w[bt] + i - bt;

            s = 0;
            for(i = 0;i < k;i++)
                s = s | (IntOne << w[i]);
            //printf("k = %d s at Makeup_eval 3    s= %d¥n",k,s);
            eval[s] = evaluate(N, k, d, w);
            goto top1;
        }
    if(bt)
        {
            goto top3;
        }
    return kk;
}
float evaluate(int N, int k, float d[][NMAX], int w[])
{
    int i,j;
    float e;

//  for(i = 0;i < N;i++)//Debug write
//      printf("Debug at evaluate : k = %d w[%d] = %d¥n",k,i,w[i]);//Debug write
.

//      printf("¥n");//Debug write.

    if((k == 0) || (k == 1))
        return 0.0;

    e = 0.0;
    for(j = 1;j < k;j++)
        for(i = 0;i < j;i++)
            e += d[w[i]][w[j]];

    return e / float(k);
}
int Subsets(int N, int k, int tbl[], int ks)
{
    int w[25], end[25];
    int kk, i, bt;
    int IntOne = 1;

    if(!((2 <= N) && (N <= 25) && (2 <= k) && (k <= N)))
        return 210;

    kk = ks;
    for(i = 0;i < k;i++)
        {
            w[i] = i;
            end[i] = N - k + 1 + i;
        }

    for(i = 0;i < k;i++)
        tbl[kk] = tbl[kk] | (IntOne << w[i]);

    top1:

    bt = k - 1;

    top2:

    kk++;
    w[bt]++;
    if(w[bt] < N)
        {
            for(i = 0;i < k;i++)
                tbl[kk] = tbl[kk] | (IntOne << w[i]);

            goto top2;
        }

    top3:

    bt--;
    w[bt]++;
    if(w[bt] < end[bt])
        {
            for(i = bt + 1;i < k;i++)
                w[i] = w[bt] + i - bt;

            for(i = 0;i < k;i++)
                tbl[kk] = tbl[kk] | (IntOne << w[i]);

            goto top1;
        }
```

```
    if(bt)
        {
            goto top3;
        }


    return kk;

}
int Prepare_Network(int M, int N, unsigned int NS[], short int min[], short int
max[])
{
// We name data mined not to use MO//
    unsigned int pscl[25 + 1], wpscl[25 + 1];
    int i, L, K;
    if(!((2 <= M) && (M <= N) && (N <= 25)))
        return 310;

    pscl[0] = pscl[1] = 1;
    for(i = 2;i <= N;i++)
        {
            for(L = 1;L < i;L++)
                wpscl[L] = pscl[L - 1] + pscl[L];

            for(L = 1;L < i;L++)
                pscl[L] = wpscl[L];

            pscl[i] = 1;
        }

    for(K = 1;K <= M;K++)
        max[K] = N - M + K;

    if(!(N % M))
        {
            for(K = 1;K <= M;K++)
                min[K] = K * (N / M);
        }
    else
        {
            for(K = 1;K <= N - M * (N / M);K++)
                min[K] = (N / M + 1) * K;

            for(K = N - (N / M) * M + 1;K <= M;K++)
                min[K] = N - (M - K) * (N / M);
        }

    for(K = 1;K <= M;K++)
        {
            NS[K] = 0;
            for(L = min[K];L <= max[K];L++)
                NS[K] += pscl[L];
        }

    return 0;
}
```